THE UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

ONLINE COLLABORATIVE TOOLS FOR ENGINEERING EDUCATION USING

SHOCKWAVE TECHNOLOGIES

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

degree of

MASTERS OF SCIENCE

By

MOHAMMAD H. ALRAMAHI
Norman, Oklahoma
2003
ONLINE COLLABORATIVE TOOLS FOR ENGINEERING EDUCATION USING SHOCKWAVE TECHNOLOGIES

A THESIS APPROVED FOR THE

SCHOOL OF AEROSPACE AND MECHANICAL ENGINEERING

BY

Chair: ________________________________

Dr. Kurt Gramoll

Member: ______________________________

Dr. Zahed Siddique

Member: ______________________________

Dr. Feng-Chyuan Lai
Acknowledgement

“Proclaim! And thy Lord is Most Bountiful. He Who taught (the use of) the pen. Taught man that which he knew not.” [Quran 96:3-5].

I would like to take this opportunity to express my deepest and sincere appreciation to my advisor Dr. Kurt Gramoll for the endless support and encouragement he provided me throughout the duration of my studies. I would like also to take the chance to thank the faculty and staff of the Mechanical Engineering department at the University of Oklahoma for their kindness and cooperation. I also wish to extend my appreciation to the staff of International Student Office and Center for Student Life for all their help and the good times we had together that made my life in the US a little easier.

For the past two years, there are a few people whom their existence in my life, and their continuous support and unbeatable generosity always was a source of inspiration to me. I would like to thank my Lebanese roommate, AbedulRahman, for his kindness and patience, my little sister Jessi for her support, inspiration and for taking good care of me and covering me at night, Abu Jamil and his family for making my life easy, my friend Eric and his wife Ashley for their valuable friendship, and the members of the Islamic Society of Norman for being my second family.

I would like to express my sincere thanks to my aunt Sawsan and her family in Canada for always keeping in touch with me, and providing me with advices when I needed them. Not to forget my best friend Steven and his wife Diana for the good times we had together and the activities we organized, and Stefanie for all the support she provided me and my causes with.
There is one person in my life I will never know how to thank. Someone was and is always there for me, even if physically we were far apart from each other, but our souls never separate. Thank you Kathleen for your precious friendship, and for being there for me when I was in needed for someone to talk to.

Most of all, and above all, a very special thank you to my mom and dad, who spent their best days in their lives to give to me, and still I feel all the need to you and your presence beside me. I wouldn’t be here today without you, and I would never be. All my success is a result for the days and nights you spent on me, and no words in this world are enough to show my appreciation. I will not forget to thank my family, teachers, and friend back home for standing behind me for all those years. Hoping that this paper benefits the people of my beloved homes, Palestine and Jordan, as well as my host country, USA.
# Table of Contents

Acknowledgement ........................................................................................................ iv  
List of Tables ............................................................................................................. ix  
List of Illustrations .................................................................................................... x  

Abstract ....................................................................................................................... xii  

Chapter One: Online Learning .................................................................................. 1  
1.1 Rich Internet Applications .................................................................................. 1  
1.2 Producing Lectures for the WWW ...................................................................... 2  
1.3 Software Engagement into Online Learning ..................................................... 5  

Chapter Two: Literature Survey .............................................................................. 7  
2.1 Lecture Creation Tools ..................................................................................... 7  
2.2 Web Collaboration is the Answer ...................................................................... 8  
2.3 Drawing Boards for Lecture Creation ............................................................... 9  
2.3.1 EML Lecture Creation Board ...................................................................... 9  
2.3.2 NCSU Tool .................................................................................................. 11  
2.3.3 Imagination at Work Drawing Board .......................................................... 11  
2.3.4 Centra Symposium ....................................................................................... 13  

Chapter Three: Authoring Tools ......................................................................... 14  
3.1 Macromedia Contribution in online learning .................................................... 15  
3.2 Client-Side Authoring Tools ............................................................................ 16  
3.2.1 Macromedia Dreamweaver ........................................................................ 16
Chapter Four: Fundamentals of Engineering Exam Tools

4.1 Logical Design and Microprocessor Architecture Chapters

4.2 Numerical Analysis Methods Chapter

4.3 Targeted Platforms

4.4 Development Tools

Chapter Five: Shear-Moment Diagram Test Tool

5.1 Tool Overview

5.2 Targeted Platforms

5.3 Development Tools

Chapter Six: Online Collaborative ‘LectureBoard’ Tool

6.1 LectureBoard as a Collaboration Tool

6.1.1 Interface

6.1.2 Privilege Levels

6.1.3 Usage

6.2 Design Architecture
List of Tables

Table 2.1: Most frequently used tools for creating computer-based training applications 8

Table 3.1: Macromedia Dreamweaver features 16

Table 3.2: Macromedia Flash features 18

Table 3.3: Macromedia Director features 20

Table 3.4: Macromedia Flash Communication Server features 22

Table 3.5: ASP Flash Turbine features 25

Table 4.1: Different numbering systems and the digits used 32
List of Illustrations

**Figure 1.1:** Connectivity between HTML pages and server 5

**Figure 2.1:** Snapshot of EML lecture creation board 9

**Figure 2.2:** Snapshot of Imagination At Work drawing board 12

**Figure 3.1:** Snapshot shows Dreamweaver MX main window elements 17

**Figure 3.2:** Snapshot shows Flash MX main window elements 18

**Figure 3.3:** Snapshot shows Director’s main window elements 20

**Figure 3.4:** FCS works under IIS in order to connect several Flash 6 player clients together 23

**Figure 3.5:** IIS and ASP connectivity architecture 24

**Figure 3.6:** Connection between ASP Turbine and the media elements used to dynamically generate a shockwave Flash movie 26

**Figure 4.1:** Snapshot of Electrical Engineering afternoon FE test 29

**Figure 4.2:** Conversion calculator 30

**Figure 4.3:** Two examples of how to use the conversion calculator 31

**Figure 4.4:** Conversion calculator shows an error in the input 32

**Figure 4.5:** Numerical integration calculator 33

**Figure 5.1:** The problem description screen of the shear-moment diagram tool 37

**Figure 5.2:** Sections positions at 3 and 6 38

**Figure 5.3:** The resultant graph of the three equations 39

**Figure 5.4:** [Left] A snapshot of the answer-view with student’s correct answer [Right] The student’s wrong answer 40
| Figure 6.1: | Snapshot of the LectureBoard shows the main elements | 43 |
| Figure 6.2: | Snapshot of a sample EMT file | 48 |
| Figure 6.3: | The architecture diagram of the LectureBoard | 49 |
| Figure 6.4: | A sample EMT file with out-of-order commands numbers | 51 |
Abstract

Distance learning has become a major issue in our modern life. Although many tools and software have been developed through the years, improvements and new inventions are still needed. One area in particular that needs more investigation is web-based collaboration.

The term collaboration is not new to the web-based systems, and recently, a number of server and client technologies have been developed to enable people to work together regardless of the physical location or time difference among them. In education, and with the innovation of online learning, interactive collaboration has become critical to assist both the instructors and the students in working together on lectures and solving problems from a distance.

This thesis researches possible usages of shockwave technologies to find strategic solutions for the collaboration problem in the online learning world, and to develop various tools for “eCourses” engineering course system, at the University of Oklahoma. For example, a tool that enables the creation of web-ready Flash lectures online in collaboration with the students in the class was developed. Flash shockwave movies are considered important for building online courses and tools, as well as for the research of this thesis. Flash shockwave movies have small file sizes, the player exists in almost all computers connected to the web, and the Macromedia Inc. continues to enhance and add to the shockwave technology.

The tools developed in this research have been integrated with the eCourses learning system. The first tool is a website for the Fundamental of Engineering (FE) afternoon exam for Electrical engineer. This website provides the student with an easy
access to a thorough and well-explained review for the FE exam. It consists of three sections: theory, examples and problems. The theory section of the website is a detailed review for the theories of a certain subject. It is also provided with simulations that help making the understanding of some hard parts easier. Few examples are introduced to the user in order to demonstrate the application of the theory explained in the theory section. Some assessment problems are provided to the user in the problems section.

The second tool was developed for submitting the shear-moment diagram problems solutions for the Mechanics of Materials course. Currently, the eCourses system has a testing mechanism that relies on multiple choices. For the shear-moment diagram problems, the interest is paid towards the equations and the final graph, which is hard to be integrated within the current testing system. The development of this tool came to solve this issue and enable the student to submit the diagram equations and graph along with the rest of the answers for the test problems.

To ease the online communication between the instructor and the student, and to provide a tool for online, collaborative and spontaneous lecture creation, a program called LectureBoard has been developed. The LectureBoard is a collaborative, vector-based, Flash-based drawing board. It is also provided with voice and text chat. Moreover, the LectureBoard is capable of recoding the session and save it in Flash shockwave format for later playback. The LectureBoard is integrated with the eCourses system, and can be accessed by the registered users.
Chapter One: Online Learning

The World Wide Web (WWW) has become a key medium for education and provides new modes for information delivery. However, the issue of what is the best way to implement it in the classroom still arises. Building static web pages with hyperlinks, bitmap images, diagrams, and videos have been used for a number of years. New methods to enhance online learning have been introduced, such as collaboration. New technologies have appeared to satisfy the needs of the educational institutes, as well as the instructors and the students [1].

1.1 Rich Internet Applications

In order to provide the web user a more effective and intuitive Internet experience, rich Internet applications are needed. Such applications result from combining the user’s experience of desktop applications with the flexibility of traditional web applications. Combining both experiences makes the creation of interactive and rich Internet applications possible, which can be delivered to the users through standard HTTP. The basic architecture of a typical Internet application includes a client-side application that connects to server back-end through various client/server technologies. This kind of architecture can be secure, scalable, and well suited for the new service-oriented model being driven by the adoption of web services.

One of the key factors for the success of rich Internet applications is to have them run within the Web browsers, along with HTML pages. Because of the ability to
run over the web, these applications will have the advantage of being cross-platform ones, which will extend the audience base to the largest possible [2].

1.2 Producing Lectures for the WWW

Instructors follow a variety of education techniques to deliver information to their students either in the classroom or from a distance. Lately, there is an increasing interest in distance learning and online lectures that can be referred to the increasing popularity of the Internet [3] that provides versatile, cost and time effective solutions for distance learning through online lectures.

A common way of producing lectures for online use is through PowerPoint presentations. PowerPoint is popular for producing lectures, showing them, and saving them for online playback. Although the presentation file size is relatively small, it is still considered too large for the web. Moreover, PowerPoint slides method offers limited interactivity for the user. Also, in order to play a presentation online, the user has to have Microsoft PowerPoint® installed on the local machine. Furthermore, it lacks one important concept that has been introduced lately in the online learning, collaboration. It is more like the instructor builds the slides and the student turns them over later on.

There have been several software packages developed in order to tackle the lecture creation for online usage, and the collaboration issues. One of the well known, and maybe most widely used, software available in the market is called Centra®[4]. Centra provides virtual classroom solution through a product called Symposium. This application has many useful features such as audio/video conferencing, whiteboard, peer-to-peer interaction and more. However, Symposium has two parts, the client part
that the user must install on the local machine and the server side that is installed on the server. Also Symposium is platform dependent, which reduces the audience to the users of that platform. The whiteboard used in Symposium is not vector-based, which makes it hard to edit or move that drawings around. Furthermore, the high costs of purchasing, running, and maintaining that application is another problem. Finally, the Symposium is a tool developed for general use, and not optimized to be used for technical topics like engineering.

Net Meeting from Microsoft is a real-time collaboration and conferencing, Windows-based client [5]. It is supported with many features such as audio/video conferencing, collaborative whiteboard, file transfer, and program and remote desktop sharing. In spite of these features, Net Meeting must be installed on the user’s local machine that runs Windows® as its operating system. Although the provided whiteboard is able to save various pages of drawings, these pages are not saved in web-ready format, the drawings are not in a vector-based, and the voices are not recorded along with these drawings.

Another solution to facilitate online collaboration is through the use of server communication programs. In the Engineering Media Lab (EML) at the University of Oklahoma, the developers had developed several collaboration tools for online usage. One commonly used tool in the eCourses system is called the Vector Board [6]. This tool has been built with Macromedia Director, and it uses the Macromedia Multi-user Server program to coordinate online collaboration. The Vector Board provides the user with the free-hand, vector-based drawing capability. One important feature of vector-
based drawing is that the drawing maintains the same resolution by resizing. Unlike paint drawing that loses resolution when resized.

The major disadvantage of the Vector Board is the fact that it is not supported with the ability to save the drawings, as well as the sounds for later playback. The main idea of an online lecture creation tools is to provide the instructor with the ability to communicate interactively with the students online and save the results of this interaction for later use. Another disadvantage is that although Director is a popular program for web development, the shockwave player that plays Director’s files is not yet common for all Internet users.

Other tools have been developed by researchers in EML lab that use the server communication technologies to provide collaboration solutions for the online learning. For example, the Multi-user Truss Solver program allows students to collaboratively work on building system of trusses. The Solver detects any errors then works on solving the system for displacements [7]. Macromedia Director was used to develop the Truss Solver, and the Multi-user communication server was used for the sake of collaboration. Clearly, the purpose of this tool is to work on trusses problems only, and cannot be used to create online lectures.

At North Carolina State University, a new program has been developed called Belshazzar [8]. Belshazzar is client-side application that helps create lectures on a local computer by using a Cross Pad as the input and saves the drawings and sounds as a Flash movie. The use of the Flash file format is a smart idea because Flash player is installed with Internet Explorer. The flash player is common and almost every Internet user has the latest player installed.
Belshazzar is a good tool for lecture creation, but the software suffers from some drawbacks. One of the major drawbacks of this software is the lack of collaboration. Belshazzar is client-side software that does not engage with any other user online. So only the instructor constructs the lecture without student interaction.

1.3 Software Engagement into Online Learning

In the initial days of Internet development, static web pages were used for online courses. The Hyper Text Markup Language, or what is known widely as HTML, was used. HTML is simple and easy to learn and use. The Internet industry moved from boring static HTML pages to dynamic and interactive pages, through the use of JavaScript, multimedia plug-ins (such as shockwave Flash and Director players plug-ins), and the connectivity with servers via server-side applications, as the diagram below shows [9].

![Connectivity between HTML pages and server](image)

Figure 1.1: Connectivity between HTML pages and server
The research of this work made use of shockwave plug-ins to build applications and connect them to the server for further processing. Chapter four discusses developing a website equipped with online tools to help the students prepare for the Electrical Engineering portion of the Fundamental of Engineering exam. Chapter five explores a new tool developed for the Mechanics of Material online course. This tool helps the students draw shear-moment diagrams for their exam questions in order to gain partial credits, and view their drawings along with the correct answer afterwards. Director shockwave has been used to build the tools discussed in those chapters.

Chapter six discusses the development of a collaboration tool, LectureBoard, to solve the problems that faced the packages and tools discussed in the previous section. LectureBoard enables the instructor and the students to collaboratively build lectures online, and save them in shockwave Flash format (SWF format) for later playback. Moreover, this tool has been integrated with the eCourses system developed at EML. Macromedia Flash MX technology has been used to build and run the tool.
Chapter Two: Literature Survey

This thesis introduces three tools for online learning: a website that provides a review for the afternoon Fundamental of Engineering exam for Electrical Engineers, a shear-moment diagram answer submission tool for the Mechanics of Materials, and an online collaborative lecture creation tool, LectureBoard, which is a flash movie that is played on the web, integrated with the eCourses system.

Not only has the interest in online learning increased with the growing of the Internet, but the interest in collaboration work has grown when developing Internet rich applications. This chapter discusses various development tools used to implement online applications, and illustrates a few applications used to build lectures for online collaboration.

2.1 Lecture Creation Tools

Unlike what most people think, Microsoft PowerPoint is not the first and most popular program used for the purpose of lecture creation. According to a recent survey conducted by Bersin & Associates in March 2003, PowerPoint occupies second place. The table below shows the results of the survey [10]. The respondents were allowed to select more than one tool; therefore the percentages total more than 100%.
Table 2.1: Most frequently used tools for creating computer-based training applications

<table>
<thead>
<tr>
<th>Authoring tool</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dreamweaver</td>
<td>52%</td>
</tr>
<tr>
<td>PowerPoint</td>
<td>48%</td>
</tr>
<tr>
<td>Flash</td>
<td>46%</td>
</tr>
<tr>
<td>Word</td>
<td>22%</td>
</tr>
<tr>
<td>FrontPage</td>
<td>21%</td>
</tr>
<tr>
<td>Authorware</td>
<td>20%</td>
</tr>
</tbody>
</table>

The survey shows that lecture builders tend to use Dreamweaver to produce lectures that are easily deployed on the web. Macromedia Flash is also showing a growing reputation in the online learning world.

2.2 Web Collaboration is the Answer

The interest in collaboration is not new to the Internet world. It was commercially introduced in 1996. One of the major factors that has helped its rapid use is that 87% of U.S. workers have broadband access [11]. The collaboration horizon did not stop at companies, but also expanded to include the education section. Many people are investigating possible uses for collaboration to enhance the online learning methods they are using [12].

One of the most time consuming task for an instructor is to create a lecture in a web-ready format. Although creating a lecture may seem to be individual work for the instructor, it is in fact a result of the collaboration between students and their instructor. Thus the need arises for a tool that enables the instructors and the students to interact.
with each other, collaboratively work on a lecture, and then give the option to record it for later use.

The following section discusses some of these tools and ideas, and presents the advantages and disadvantages of each one, as well as the technologies used to build such tools.

2.3 Drawing Boards for Lecture Creation

2.3.1 EML Lecture Creation Board

Previously, Mr. Peddy [13] developed a tool to create lectures and save the final work in Flash shockwave movie format. Figure 2.1 shows an image of the tool.

![Figure 2.1: Snapshot of EML lecture creation board](image)

Figure 2.1: Snapshot of EML lecture creation board
For the client side, the technology used is ActiveX component that is written in Visual Basic 6.0 and registered to run under Internet Explorer. The component traces the user’s drawings and stores them in a text file. When done, the user loads the saved file to the server by hitting the ‘load’ button, which activates the server side application. The server side application takes the uploaded file, initiates the ASP Turbine component that generates a compiled Flash shockwave movie file.

This tool lacks a number of features. Although the tool is available online, it does not allow collaboration. Only the instructor uses the tool to create a lecture without the student interaction. In addition, the tool requires the user to download and set up the ActiveX component, which only works for certain platforms and is optimized to function under Internet Explorer. Another draw back is that the sound is recorded in WAV format, which results in a large file size. This file is converted into MP3 format through a WAV-to-MP3 converter component that has to be installed and registered on the client’s local machine. After finishing the recording, it is the user’s responsibility to manually issue the loading command to upload the wav and text files to the server and generate the Flash file.

The ActiveX component that contains the board lacks the security. The user must set the security level in the web browser to low in order to run the board. By doing this, the user exposes the machine to hack attacks as long as the security level maintains the low state.
2.3.2 NCSU Tool

The North Carolina State University has developed a client-side application to create lectures on a local computer, by using a Cross Pad as the input. The application is called Belshazzar [14].

The application is not available on the web for demo or download. However, through looking at the playback recorded movies, the tool seems to give good results with smooth playback of Flash shockwave files. The sound is too close to be synchronized with the drawings. In addition, the tool captures almost every keystroke of the input device.

However, this tool suffers from certain disadvantages. First, the application is built to work on a client computer, which means that the tool is only accessible whenever and wherever that computer is accessible. Second, the sound is not yet perfectly synchronized with the playback. The user can notice a little bit of lag between the drawings playback and the sound playback. Third, the saved Flash file is generated and stored locally without any dependency on the server, which means that the user needs to manually upload and archive it. The reason for this is that the application is a separate program and is not part of a complete online system. Finally, this application lacks collaboration. The students do not take part in the process of creating a lecture.

2.3.3 Imagination at Work Drawing Board

The drawing board produced by Imagination at Work [15], is trying to provide a Flash based drawing board that has potential use in business as well as in education. The tool has a simple interface, with a drop down menu system enables the users to
change the line thickness and color, choose the grid type, clear the drawings and quickly preview what has been drawn. The mouse cursor is in the shape of a marker that changes colors based on the user’s choice. Figure 2.2 shows a snapshot of the drawing board.

![Figure 2.2: Snapshot of Imagination at Work drawing board](image)

This drawing board was developed by Flash, which means that there is no need for the user to install any special plug-ins. Moreover, the application size is small and suits the Internet. The user can preview the session by clicking on the ‘preview’ button. The preview is not stored in a file, but rather in the server’s memory. However, the tool suffers from a number of problems. First, the board lacks collaboration. The users are not able to interact or communicate with each other through this tool. Although the user
can have a preview of the session, the drawing board lacks the sense of timing between each drawing object. Also, the sound is not recorded. Finally, the drawings cannot be saved as a file for later playback.

2.3.4 Centra Symposium

The Symposium from Centra® [16] is developed to be the solution for a live online learning over a network. It enables students, groups, and instructors to interact and collaborate through the Internet. The features included in Symposium are IP audio/video conferencing, breakout rooms and labs, quizzes, whiteboard, peer-to-peer chat, and interaction and session recording. The bandwidth required is low and affordable even through dial-up.

Not much information is available on the Symposium except what is posted on their website. Since the University of Oklahoma did not purchase this software, no experiments were performed. However, reading through the manual posted online at Centra® website, the tool suffers from disadvantages and lacks some features.

Although the Symposium supports collaboration, which has not been supported by any of the previously discussed tools, this tool itself is an application that has to be installed and setup in the client’s local machine. The server used for the tool is located at Centra’s company site, and the license is expensive. Finally, the recording and playback, is saved as AVI format and is not in Flash format. This makes the movie files large and not scalable.
Chapter Three: Authoring Tools

Throughout the years, many companies have been interested in developing software applications and packages for the Web. Some of these companies realized the importance of the Internet in education and decided to develop tools that could be used to construct learning material. One of the biggest names in the online learning era, as well as the Internet development tools, is Macromedia.

Macromedia MX, which is the latest Rich Internet Applications complete development package, is designed to work together seamlessly. The Macromedia MX product family includes major new releases of existing Macromedia products, as well as new technologies and products. Throughout this paper, we discussed three major Macromedia MX products: Flash MX, Director MX, and Flash Communication Server MX [17].

Since online learning forms a whole system with server and client sides, other technologies and tools are also used. For database activities, Active Server Pages (ASP) technology is used. The ASP technology works with Internet Information Server (IIS) within Microsoft Windows® Server edition. An interesting software that adds functions to Flash is called ASP Turbine, by Blue®Pacific. The Turbine is a server side application that generates a shockwave Flash file through parsing text file containing Turbine commands. This software works under MS Windows® 2000 server, IIS 5.0 and may executed through ASP. Another technology that is used is the Components Object Modeling (COM+), which enables the usage of C++ programming language to perform certain tasks within our system. More details on these technologies and tools are discussed in this chapter.
3.1 Macromedia Contribution in online learning

Macromedia products have been widely used by million users in the education field [18]. The MX product family supports the implementation of interactive, innovative, and collaborative learning applications and tools. With all the new features added to the new Macromedia products, the MX product family, the horizon of innovative teaching is larger than ever.

The use of Macromedia MX products shortens the development time, guarantees a cross-platform production, and affords collaboration within affordable environment. Because of the ready to use templates, the development time has been shorter than any time earlier. In addition, Macromedia releases software suits that contain products integrated with each other that can be used to build an application from its beginning to its end. For example, the Macromedia online learning Suite that combines Macromedia Authorware, Macromedia Flash MX, and Dreamweaver MX in one convenient package for creating online learning applications and websites.

Flash player plug-in has been widely spread over the web browsers. Although Director player plug-in is not as famous, it can be downloaded for free from Macromedia website. Both players are available for all different kinds of platforms, from smart cell phones, Pocket PC, to regular desktop and Laptop computers with various operative systems, as well as the Tablet PC. Hence, any application written in Flash or Director is presumed to be multiplatform.

Communication is an essential element in education. Both teachers and students need to have the means to keep communicating with each other in clear and easy way. With the new Macromedia Flash Communication Server MX technology, collaboration
between instructors and teachers becomes easier, more reliable, and affordable. Building video, audio, chat, and information sharing tools has been an easy task and non-time consuming because of the ready made Macromedia communication components offered with this server.

3.2 Client-Side Authoring Tools

3.2.1 Macromedia Dreamweaver

We mentioned in the previous chapter that Macromedia Dreamweaver has become the leading application for website authoring according Bersin & Associates’ survey in March 2003. Table 3.1 shows some of the features of Macromedia Dreamweaver and those were used in the research:

Table 3.1: Macromedia Dreamweaver features

<table>
<thead>
<tr>
<th>Features</th>
<th>Used Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Easy to use interface</td>
<td>✓</td>
</tr>
<tr>
<td>2. Easy to access functionality</td>
<td>✓</td>
</tr>
<tr>
<td>3. Clean HTML generated code without the redundant extra tags</td>
<td></td>
</tr>
<tr>
<td>4. Integrity with shockwave files (Director and Flash)</td>
<td>✓</td>
</tr>
<tr>
<td>5. Integrity with Server side technologies such as ASP and Per</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.1 shows a typical page in the Dreamweaver application. Notice the tool bars at that top and to the left of the design window.
Dreamweaver was used during this research to build static HTML pages that include multimedia, forms, and Flash and Director shockwave files. Also it has been used to construct the ASP pages used for the LectureBoard project described in chapter six.

### 3.2.2 Macromedia Flash

Macromedia pioneered the web industry in making web pages look more interactive, beautiful, and faster to download through the production of Director and Flash. For instance, Flash player plug-in is supported by more than 95% of the browser market [19]. Table 3.2 shows some of the features of Flash.
Table 3.2: Macromedia Flash features

<table>
<thead>
<tr>
<th>Features</th>
<th>Used Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Vector-based animation and graphics</td>
<td>✓</td>
</tr>
<tr>
<td>2. Great compression for images and videos</td>
<td></td>
</tr>
<tr>
<td>3. Fully featured ActionScript programming language</td>
<td></td>
</tr>
<tr>
<td>4. Flash player is supported by more than 95% of the browser</td>
<td>✓</td>
</tr>
<tr>
<td>5. Flash shockwave file is cross platform [20]</td>
<td>✓</td>
</tr>
</tbody>
</table>

With Flash MX, building applications for academic is easier due in part to the intuitive interface, “getting-started” tutorials and the powerful ActionScript. The diagram below shows a typical screen of Flash MX application window.

Figure 3.2: Snapshot shows Flash MX main window elements
Although ActionScript has improved since the early versions of Flash, its API library still lacks many functions. For example, the Mathematical functions library is not yet appropriate to be used for sophisticated scientific applications.

Flash technology is used to build the LectureBoard tool. Since the drawing tool is built with Flash, only flash player is needed to start using it. That gives an impression that no matter what web browser or platform the user has, the tool will work fine. LectureBoard is featured with collaboration and saving for playback abilities.

3.2.3 Macromedia Director

The idea of Director started with Video Works from Macromind. Macromind became Macromedia later, and Video Works became Director. Although originally the production was only for Apple Macintosh computers, a player for windows made the generated shockwave file platform independent.

The application window of Director contains four main elements: stage, score, cast library, and property Inspector, as well as other tool bars. The figure below shows these elements.
Figure 3.3: Snapshot shows Director’s main window elements.

Director is one of the most powerful tools for web authoring because of its features that Macromedia keeps adding to them. Table 3.3 shows some of these features.

Table 3.3: Macromedia Director features

<table>
<thead>
<tr>
<th>Features</th>
<th>Used Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Small shockwave files size</td>
<td>✓</td>
</tr>
<tr>
<td>2. Director shockwave file is cross-platform</td>
<td></td>
</tr>
<tr>
<td>3. Extensible through 3rd party Xtras</td>
<td></td>
</tr>
<tr>
<td>4. Lingo is supported with wide library of functions and APIs such as audio and video controls, math functions, drawings and more</td>
<td>✓</td>
</tr>
<tr>
<td>5. Object-oriented programming in Lingo through behaviors</td>
<td></td>
</tr>
<tr>
<td>6. 3D engine to Director</td>
<td></td>
</tr>
</tbody>
</table>
However, with all these impressive features of Director, Flash shockwave files gained more popularity and dominated the web over Director shockwave. The reason why is simple and goes back to the days when the modems were the main media to access the web. Modems are slow, so web developers minimized the size of the produced files as much as possible. Although Director shockwave files were small in size, Flash files proved to be even smaller and in effect more practical for such slow Internet access [21]. Flash also had more vector-based drawing capabilities that attracted graphic artists.

Flash and Director are not competing with each other. Most of the time it is clear when to prefer one of them over the other. For scientific, robust, and 3D applications, Director is the choice, and is still the best development tool for CD-ROM delivery.

In this thesis, the Director has been used to develop online tools that provide the students with assistance. In Chapter four, the Director has been used to implement a calculator that converts a number with a given radix to its equivalent in the other radices. Another calculator has been implemented through Director that finds the value of a limited integration for a given function using two methods, Euler and Trapezoidal, and provides the accurate answer for the purpose of comparison. In Chapter five, Director was used to build the shear-moment diagram tool for Mechanics of Material class. The generated shockwave file reads the input, draws the problem statement, and accepts the student’s input for the problem, draws it, and then sends it to the server.
3.3 Server-Side Development Tools

3.3.1 Macromedia Flash Communication Server MX (FCS)

The second half of 2002, Macromedia released a major new component for Flash, Macromedia Flash Communication Server MX from Macromedia (FCS). Not only does this server component work for Flash applications, as the name indicates, but also works with Director. Macromedia Director MX is provided with the library to communicate and use the services offered by FCS. Because of this capability, Macromedia discontinued the Shockwave Multi-user server used previously by Director for producing collaborative, multi-user applications. Table 3.4 lists some of the FCS features and those used in this research.

Table 3.4: Macromedia Flash Communication Server features

<table>
<thead>
<tr>
<th>Features</th>
<th>Used Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Server-side technology</td>
<td>✓</td>
</tr>
<tr>
<td>2. Connects Flash 6 players</td>
<td>✓</td>
</tr>
<tr>
<td>3. Connects .NET and J2EE applications</td>
<td></td>
</tr>
<tr>
<td>4. Allows streaming audio and video over the local network and the Internet</td>
<td>✓</td>
</tr>
<tr>
<td>5. Supported with free and easy to use communication components</td>
<td>✓</td>
</tr>
</tbody>
</table>

Any application that is developed to work with Flash Communication Server MX is called a Flash Communication Application. It can be easily imagined as a network hub that connects Flash 6 players using a bidirectional protocol called Real-
Time Messaging Protocol (RTMP) [22]. The figure below shows how FCS works under IIS in order to connect several clients together.

![Diagram of FCS under IIS](image)

Figure 3.4: FCS works under IIS in order to connect several Flash 6 player clients together.

In this research, the Flash Communication Server MX is used in the LectureBoard for two main tasks: first, for interactivity and collaboration between the different connected clients; and second, to record and save the voice stream for later playback.

### 3.3.2 Active Server Pages (ASP)

ASP is a server side technology that allows two-way communication with HTML pages. A developer can create an ASP page using either VBScript or JScript language. Microsoft came up with this technology with the release of their Internet Information Server (IIS). IIS is a service that runs on the server that supports dynamic
applications written in ASP and other scripting techniques through a single DLL file named ‘asp.dll’ [23].

Once a user requests an ASP page, the DLL file parses it and then sends it to the appropriate interpreter engine that corresponds to the scripting language used in the page. The engine executes the script code by parsing the page and executing the commands line by line. The results are combined with any HTML tags found in the ASP page. When done, the interpreter engine sends the response to the IIS that passes it back to the client which originally requested the page. The ASP script can perform mathematical operations, access different databases, instantiate, and run COM server components. This process is illustrated in figure 3.5.

![Figure 3.5: IIS and ASP connectivity architecture](image-url)
In this work, ASP is used with the LectureBoard, for three main tasks: to access the database for storing and retrieving certain data; to save and manipulate text files that hold the drawing commands; and to initiate the ASP Turbine, by passing the required parameters to it, and then send the request to create a SWF with the Turbine component.

### 3.3.3 ASP Flash Turbine

ASP Flash Turbine is a server side component that generates dynamic Flash movies through ASP code. The ASP Flash Turbine is implemented as a COM component in order to offer the best possible performance on IIS web servers [24]. The ASP page can be written in several languages such as VBScript, Jscript, and even PerlScript. The application works under Windows 2000 Server edition and with Internet Information Server (IIS) versions 4.0 and 5.0. Table 3.5 lists some of the features of ASP Turbine.

<table>
<thead>
<tr>
<th>Features</th>
<th>Used Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Server-side component</td>
<td>✔</td>
</tr>
<tr>
<td>2. Dynamically generates Flash shockwave files</td>
<td>✔</td>
</tr>
<tr>
<td>3. Works under IIS 4.0 and up</td>
<td>✔</td>
</tr>
<tr>
<td>4. Initiated by ASP</td>
<td>✔</td>
</tr>
<tr>
<td>5. Uses Flash templates to generate the flash shockwave file</td>
<td>✔</td>
</tr>
</tbody>
</table>

The mechanism of how Flash Turbine works is simple. When the user requests the ASP script page that contains the code to initiate Turbine, the web server executes
the code in the page, which results in creating an instance of the Turbine component. Afterwards, Turbine accesses the required media files such as texts, images, videos and text files that include Flash Turbine commands. As a result, a shockwave Flash movie is generated, integrating the accessed content in a movie template. Figure 3.6 shows the connection between all the elements involved in the generation process.

Figure 3.6: Connection between ASP Turbine and the media elements used to dynamically generate a shockwave Flash movie

The generated Flash movie may contain text, images, tables, charts and drawings, sounds, videos, and even ActionScript scripts. Referring to figure 3.6, the creation of Flash movie using Turbine requires Flash template file. In Macromedia Flash 5, a template may be designed and saved in template format (SWT). The ASP page requests Flash Turbine to use that template and links it with all the necessary media. Then, Flash Turbine works in the background to generate the shockwave Flash
file. A list of APIs (Application Programming Interfaces) is supported by Turbine to enable the users to create a customizable Flash shockwave files that correspond to users inputs.

The ASP Turbine is used in the LectureBoard, which is described in more detail in chapter six. The main purpose of using it is to generate a playback Flash file that contains all the drawings submitted by the users, along with the voice conversation that took place during the recording of the session.

It is worth mentioning that the ASP generated Flash movie may be played back, but cannot be edited or changed. In other words, it generates an SWF file, not an FLA file, which is the source code format for Flash.

### 3.3.4 Component Object Model (COM+) Technology

COM is an object-oriented programming architecture, combined with a set of run-time services, called COM library. An application created to fit in COM specifications is reusable, scalable and accessible by numerous languages [25].

The benefits of COM lie in the fact that it is language independent. Developers can usually create COM components using any Object-Oriented programming language, such as C++, Java, and Visual Basic. The result is a binary file in the form of DLL or EXE.

In this research, COM has been used in the LectureBoard tool. It is used to parse text files written in certain formats and generate a final polished text file ready to be used by ASP Turbine. It runs on the server and does not to be downloaded by the user.
Chapter Four: Fundamentals of Engineering Exam Tools

The Fundamentals of Engineering exam, or FE for short, is required by states to become a licensed engineer and thus eligible to perform certain engineering work. Since the fall of 1996 the exam has undergone changes. It has become two parts, and administrated in the morning and one in the afternoon. The morning section is about general information in various topics like math, chemistry, physics, and other general courses. The afternoon portion is concentrated in the core courses of the major [26], ie. Civil, mechanical, industrial, electrical, and chemical.

Many available books and tutorials are available to students for a reviewing but generally there is a cost. Students usually like to have certain tools that help them during their study progress. While studying, certain types of questions come to their minds, questions like “what if I changed this to ..,” or “what will be the solution if I do that…”

This chapter is devoted to tools which have been developed for the Electrical Engineering afternoon FE exam. The parts that have been taken care of in this paper are related to the computer architecture design courses in addition to the numerical analysis course.

Each topic in the FE website consists of three sections: theory, examples and problems section. The theory section includes a review of the theories included in the exam. Simulations are used to demonstrate the theories. The examples section includes several samples that show the application of the theories presented in the theory section. Finally, the problems section provides a few exercises in order to measure the students
understanding to the theories and their ability to apply them after reviewing the examples.

The navigation from one section of the website to another, and from one chapter of the FE to another is made easy through simple, text-based navigation menus, as shown in figure 4.1 below.

Figure 4.1: Snapshot of Electrical Engineering afternoon FE test

The website presented in this section can be examined at http://eml.ou.edu. It is offered to the audience free of charge, and hosted by the Engineering Media Lab (EML) servers.

4.1 Logical Design and Microprocessor Architecture Chapters

Boolean operations and logical design are the first computer hardware related topics students learn. Humans use the decimal numbering system, which uses the digits 0 – 9. Unfortunately the computer hardware does not understand this numbering system - it uses the binary one, which contains only 0 and 1. Other numbering systems have
been developed through the years and been used when dealing with the computers, such as octal and hexadecimal systems. The conversion from one system to another is not always straightforward procedure, and may seem troublesome to students.

In the FE website (after explaining the conversion algorithm) the students are provided with a tool called a conversion calculator. This calculator converts a number from one numbering system to another, see figure 4.2. This tool can be used in several ways. It can be used to check the student’s answer on a problem, or to provide an answer for questions that are developed in the student’s mind.

![Conversion Calculator](image)

**Figure 4.2: Conversion calculator**

The conversion calculator enables the student to enter a number, choose the radix, and then get the result of the conversion to the radices shown in the result panel to the left, as shown in figure 4.3
The student chooses the radix of the input number by moving the slider in the sliding bar. A common mistake the student may make at this point is to input a number that is not recognized by the chosen radix system. For example, the number 201 is fine to be entered for all the radices except the binary, since the binary system uses only two digits: 0 and 1, and the number contains the digit ‘2’. Generally, if a number contains digit(s) that are not part of those used for a numbering system, it will not be recognized by that system, and will generate an as shown in figure 4.4.
Different numbering systems use different digits to represent numbers. Table 4.1 shows the commonly known numbering systems and the digits used in each one.

Table 4.1: Different numbering systems and the digits used

<table>
<thead>
<tr>
<th>Numbering System</th>
<th>Radix</th>
<th>Digits</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary</td>
<td>2</td>
<td>0,1</td>
<td>100101</td>
</tr>
<tr>
<td>Octal</td>
<td>8</td>
<td>0,1,2,3,4,5,6,7</td>
<td>572</td>
</tr>
<tr>
<td>Decimal</td>
<td>10</td>
<td>0,1,2,3,4,5,6,7,8,9</td>
<td>193</td>
</tr>
<tr>
<td>Hexadecimal</td>
<td>16</td>
<td>0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F</td>
<td>2A7</td>
</tr>
</tbody>
</table>

4.2 Numerical Analysis Methods Chapter

Numerical analysis plays a key role in engineering computations. One important subject is to find the approximate value of a limited integration using various numerical methods. Two main methods are used and discussed in this section to find the result of an integration numerically: Euler’s method and Trapezoidal method. In order to provide the student with more understanding of the differences between these two methods, and to compare the exact solutions with the approximate ones, a numerical integration calculator was developed and added as part of the website (see figure 4.5).
The numerical integration calculator is able to solve three types of functions: exponential, sinusoidal, and polynomial of $3^{rd}$ degree. The solution is given as a result of Euler and Trapezoidal methods. For the sake of comparison of accuracy between the two methods, the exact answer is also provided.

This calculator is self-explanatory. All what the student has to do is to choose the target function, input the necessary factors, and provide the calculator with the integration interval - i.e, initial and final points - and then read the results.

Since the number of subintervals used in both Euler and Trapezoidal methods play a role in determining the accuracy of the answer, the calculator has been equipped with a sliding bar to choose the number of subintervals. As the slider moves, the calculator redoes the mathematical calculations and shows the results immediately.
providing the results of the two methods, as well as the exact answer, the student clearly sees the difference in accuracy between Euler and Trapezoidal methods. Also the student demonstrates the fact that the larger number of subintervals used, the better the numerical methods converge to the exact value.

### 4.3 Targeted Platforms

The main targeted platform was Microsoft Windows®, such as 98, 2000, and XP. The preliminary tests for the website design were performed on the Internet Explorer web browser. Another operating system, Mac OS®, was used for secondary tests in order to make sure that the website maintains the same design. Generally, the design remained almost the same and no changes were needed. However, in certain cases some changes needed to be done in order for the web page to be perfectly viewable on Mac machines. Minor changes are fixed and tested on both platforms. However, the major ones on both platforms were dealt with on Windows® platform.

### 4.4 Development Tools

Since the platforms were determined earlier, the tools were chosen that are capable of performing the required task on both platforms.

In the development of the tools described in this chapter, two main Macromedia products were used. The first product is Macromedia Dreamweaver, which was used to build the static HTML pages for the website. It makes it easier to integrate the produced shockwave files with these pages. The second tool is Macromedia Director 8.5. The
The reason behind choosing Director over Flash is that the mathematical capabilities of Director exceed those of Flash. While Flash ActionScript gains more power with the new versions, it still lacks the power of mathematics. Unlike ActionScript, Lingo programming scripts are equipped with a full library of mathematical functions, which are capable of performing complex mathematical operations.
Chapter Five: Shear-Moment Diagram Test Tool

The purpose of the shear-moment diagram test tool is to enable the student to build and submit shear-moment diagrams for tests and homework assignments in order to gain potential partial credit. Building a shear-moment diagram is an important goal for the Mechanics of Materials class. Previously, eCourses test and homework system was based on only a multiple choice format, a system that does not reward the student with partial credit. Hence, this prototype tool was developed to investigate this issue.

The logic behind this tool is fairly simple but depends on the equations that a student develops through solving the problem. The student finds out the shear or moment equations, plugs in the numbers in the tool and draws the resultant diagram. If satisfied, the student submits it, whereby the equations get stored in the server for partial credit grading. The student can review the submitted answers, update them, and resubmit them any time during the test. After the correct answers are released, the student views his/her answer versus the correct answer.

5.1 Tool Overview

The shear-moment diagram tool was built to be integrated with the eCourses course management system. Whenever the tool is needed, a link is provided to the students for each problem. When the students click on a link, the server receives a request to initiate the tool along with the problem number. The problem specifications are taken out of the database and sent to the shockwave file. When the tool screen appears to the student, it shows a bar with the joints, forces, and moments as described
in the problem statement. An assumption made was that the highest level of complexity for the forces will be a distributed load with a triangular shape, which results in quadratic equation for the shear and cubic equation for the moment equations.

For example, assume a bar with length 9 that is connected with a pin joint at the first end, a roller joint at 6, and acted upon with a counter-clockwise moment at 3 with magnitude 6, and a force at 9 with magnitude 6. The screen will appear to the students as shown in figure 5.1 below.

![Figure 5.1: The problem description screen of the shear-moment diagram tool](image)

The student determines the number of sections the problem will be divided into, based on the understanding of the problem. The next step is typing in the equations and
drawing them for each section. In the previous example, there exist three sections. The dividers are located at 3 and 6 on the bar, as shown by the figure 5.2.

Figure 5.2: Sections positions at 3 and 6

Since this tool is an aid for the quizzes and tests, it assumes that the student works out the problem, then uses this tool to gain partial credit through submitting the graph. Since the example above is about a shear diagram with 3 sections, 3 quadratic equations are needed, one for each section. When the student feels satisfied with the resultant equations, he/she can fill in the numbers and clicks the ‘draw’ button. Then a diagram of the student’s equations, as shown by figure 5.3, will appear. The student’s answers are then sent and stored on the student’s test profile at the server when clicking the submit button.
After the answer key is released, a similar tool is used to view the student’s answer vs. the correct one. The correct answer-view tool is only activated when the exam is over and the instructor releases the correct answers. Figure 5.4 shows a snapshot of the answer-view tool.
5.2 Targeted Platforms

The targeted platform was Windows since students at the University of Oklahoma who take the class are required to have Windows operated laptops. Generally speaking, any desktops, laptops, or Tablet PC that run Microsoft Windows operating system, and has Director shockwave player 8.5 plug-in installed can use it. Although Pocket PC 2002 runs Windows operating system, a shockwave player has not been developed yet for it.

5.3 Development Tools

The only product used for the development of the shear-moment diagram test tool is Macromedia Director, because of the mathematical calculation power of its
Lingo programming language, and its vector graphing abilities. Lingo programming language makes it easy to perform the necessary mathematical operations in order to graph the equations, check data validity, and submit the results. One pre-assumption is that if users are using eCourses, then the Director shockwave plug-in is already installed on their machines.
Chapter Six: Online Collaborative ‘LectureBoard’ Tool

Web-based distance learning is in part to the growth of the Internet. There are many interactive, media-rich websites to enrich the online learning experience. However, more attention needs to be paid on a key feature for effective online learning, collaboration between the instructor and the students.

In chapter two, various tools were discussed that have been built for the purpose of collaboration. This chapter details a tool has been produced for online collaborative creation of lectures for online use. The tool is called ‘LectureBoard.’ It also explains how the LectureBoard is able to solve the disadvantages other similar tools.

6.1 LectureBoard as a Collaboration Tool

LectureBoard is a tool, designed with Macromedia Flash MX, that interacts with server-side applications and services, to enable the instructors and the students to hold an online lecture. Both instructors and students can voice chat with each other, draw on the whiteboard, and record the session for later playback. This section outlines in more details the LectureBoard interface, usage, and benefits.

6.1.1 Interface

The LectureBoard was designed to maximize the space of a common XGA resolution screen at 1024x768 pixels because it is common and gives the instructor the widest drawing board area possible. It contains a white drawing area, list of logged-in students and instructors, text chat window, and two drop-down menus to choose the line
color and thickness. The instructor’s window contains the same elements plus lecture-recording controllers (see figure 6.1).

Figure 6.1: Snapshot of the LectureBoard shows the main elements

The drawing tool menu shown in the figure 6.1 it is designed to be floating so it does not occupy space on the drawing board. The user can move it to any place in the screen at any point of time. Drawing shapes are provided within this menu bar such as line, rectangle, circle, arrow, free-hand, eraser, and object selector. It is worth mentioning that the eraser works more like a white marker in its functionality in that “erased” objects remain, though invisible against the white background.
Users of LectureBoard can communicate with each other in three different ways. First, by drawing and sharing the drawings between each other. Second, through the voice chat component integrated with this tool. All the logged-in users can voice chat and hear each other clearly. And finally they can communicate through the text chat component located at the bottom of the drawing board.

In order to make the drawings clearer and sometimes more joyful, colors and line thicknesses can be chosen for each drawing object. Moreover, for engineering users who like having accurate drawings, grids are provided.

Since it is only the privilege of the instructor to record a lecture, or a session, the recording controls are shown on his/her window only. The controls contain: an input text field for the lecture file name; two buttons, one for record/stop action, and the other to clear the drawing board; and finally a dynamic text field that shows the name of the user who owns the last drawing object on the screen.

6.1.2 Privilege Levels

The LectureBoard has been designed with two access levels in mind: students and instructors. A student user can freely use the drawing board to share drawings with other logged-in users, and may communicate with them through the use of the voice and text chat. However, students do not have the required privilege to record the session, thus the session recording controls will not be visible in the student mode. In the instructor mode, the instructor user can do all that the student user can do, plus they have the ability to clear the board and record the session.
6.1.3 Usage

There are three possible uses for the LectureBoard. The first one is as online lecture creation tool. The second is an online office hours meeting tool without recording, and last, usage as an online office hours meeting with ability to record the session.

As for the first usage, LectureBoard enables the instructor to record the online lecture, while working in collaboration with the students to build it. Since it is integrated with the eCourses system built by the developers at the Engineering Media Lab (EML), the created lecture is archived in its associated chapter and section.

This tool can also be used by the instructor and the students to meet online for an office hour session. In this case, no recording is required, and both students and instructors log in automatically in the student mode that does not support recording. Since no recording will be allowed, the archiving problem is not an issue.

The third possible use of the LectureBoard is to record online office hours sessions. There is not any difference in the interfacing and functionality than when used to create online lectures except in the archiving process. Since eCourses system is not ready yet to record and archive office hours, this idea is considered a feature for potential use sometime in the future.

6.2 Design Architecture

6.2.1 Architecture

The LectureBoard consists of both client and server sides. The client side is compiled as a Flash shockwave file and runs inside a web browsers. It is able to
communicate with the server through the server APIs supported by the ActionScript. These APIs are able to open two-ways connection channel with the server, call ASP pages and server-side ActionScript APIs, and execute a server request to the client. The client side can be looked at as a fully functional interface that sends commands to the server.

On the other side, the server handles all commands and requests received from the client. During the non-recording mode, which can also be called the student mode, the client Flash movie opens a two-ways communication channel with the Flash Communication Server MX installed on the server. When the connection is successful, the client movie keeps sending the drawing commands to the communication server, which in turn broadcasts it to all the client movies connected to it.

In detail, when a user draws an object on his/her client machine, the Flash movie calls a function developed inside a file named “main.asa” and passes the drawn object parameters to it. The “main.asa” is a text file that contains functions written using server side ActionScript APIs. These methods are interpreted and executed by Flash Communication Server MX. Once the broadcast method receives the request from the client, it saves it in a history array, and then request a function built in all the client movies passing the object parameters. The function in the client receives the parameters, creates an empty clip, and draws the requested object in it.

When a student logs into a class after it begins, he/she loses what the professor and other students stated previously, as well as anything erased from the board. The same scenario applies on the LectureBoard. When a user logs in after the start of a session, and drawings are already on the board, he/she will definitely lose any voice or
text chatting that has been made in his/her absence. This problem was resolved through programming the communication server to update the Flash movie with whatever has been drawn on the whiteboard, exactly as a real classroom. This program contains a history array stored by the server for each open session. Whenever a user logs in, the program commands the communication server to send whatever is inside the history array to the client Flash movie, calling an update function built in that movie that updates the user with the contents of the whiteboard.

As for the voice chat, it is a pre-made component, developed by Macromedia team that can be downloaded for free from the Macromedia website. This component uses the open communication channel to transfer or stream the voice over the network. For every connected user, Flash Communication Server MX creates a streaming file on the server for the streamed voice from that user. The file extension is FLV.

Recording a lecture is a more complicated process, and involves other technologies in addition to that what has been previously explained before. Since the voice is handled through the communication server, the recording of the voice also goes through it. The Flash Communication Server has the ability to record the voice stream and save it in the FLV file format.

When the instructor issues the start recording command, the client movie sends any drawing command issued by the instructor or received from any of the users, to an ASP file. The main purpose of this ASP file is to accept a string containing drawing command details, opening a text file on the server and storing the received command. The commands received by the ASP file are written in a way that contains the object number, drawing command, and the object details. This format has been referred to as
Engineering Media Lab Template, and the text file is given the EMT extension. The following figure shows a sample file.

![Sample EMT File](image)

Figure 6.2: Snapshot of a sample EMT file

Once the instructor is done with the session recording, the Flash movie issues a command to another ASP file and starts the process of creating a Flash movie for playback. This page instantiates a COM component. The component reads the EMT file, parses it, and generates a temporary text file that contains the drawing commands in order and in Turbine API format.

After the COM component is done with its task, the control goes back again to the ASP page. It saves the lecture information, such as the title and the file name, in
Microsoft Access database, and ignites the ASP turbine component to generate the required SWF file, and then deletes the temporary text file.

The ASP turbine reads a Flash 5 template and the temporary text file, parses and executes the drawing commands, and attaches them with the template to generate a compiled SWF file. Figure 6.3 shows the described design process.

![Figure 6.3: The architecture diagram of the ‘LectureBoard’](image)

The final result of the recording process is basically three files: an EMT file that is kept for future use, an FLV file that contains the voice recording, and the SWF containing the drawings recording. Since the sound and the drawings have been
recorded in two separate files, the play back webpage contains two separate flash files. One is used to play the FLV file, and the other is the generated SWF file itself.

### 6.2.2 Obstacles and Solutions

As with any program, there are numerous developmental problems and issues. Throughout this research and development process, many obstacles were faced and solved. Probably the first one worth mentioning was the lack of resources about Flash Communication Server MX. When starting with the project in June 2002, the only available reference for it was its manuals. However, the first half of next year witnessed the publication of many books regarding software.

When a client Flash movie sends a drawing command, the Flash server receives it, and then broadcasts it to all the connected clients, including the caller itself, which results in duplicated drawing at the caller’s side. To avoid the duplication, the caller deletes the original drawn object, and waits for the draw command to be received from the server in order to draw it again.

An interesting, and odd, problem was faced during the recording of the session. As mentioned earlier, when in the recording mode, the instructor’s client movie sends the drawing commands, issued by the user or by the others, to the ASP file in order to store them in a text file. Although the commands are sent in order, the ASP does not get them in the same order. This problem causes all of the drawings to be messy and sometimes makes it hard to understand the correct drawing. Figure 6.4 shows a snapshot of a file with mixed order. Notice how the command numbers at the first column are out of order.
This problem was solved by storing the command number as received from the Flash movie into the EMT file. This initiates the COM component to parse the file, sort the commands, and save the final version in a temporary text file.

Talking about the COM component, it has an additional functionality other than sorting the commands in the EMT file. All the tools that create lectures in Flash format suffer from a serious problem with the synchronization between the sound and the drawings playback. The user notices that the sound and the drawings are both off from each other by a few milliseconds, which can increases to a few seconds depending on the length of the movie. After several trials, one good way to minimize this gap is by using real time stamps. From one drawing to the next, the Flash movie calculates the
time taken, and sends it to the ASP file to store it in the EMT file along with the other
commands. Since Flash is frame-based instead of real-time, the COM component reads
the time stamp, and converts it to series of ‘Show Frame’ commands depending on the
frame rate (frames per second). The frame rate was fixed to 5 frames per second. For
example, if the time command is ‘Time 2,’ the COM will replace it with (5 x 2 = 10)
‘Show Frame’ commands.

Another reason for a lag between the drawing and the sound comes back to the
fact that two Flash movies are used for the playback. One plays the drawings and the
other plays the sound. Although it would be more efficient if both files were stored in
one file, this was not possible with the given technology. The sound file is stored in
FLV format, which is a sound format supported only by Flash MX and later versions.
Unfortunately, ASP turbine that generates the other file supports templates made by
Flash version 5 only. There wasn’t a way with the given tools and technologies by the
time of the development to combine both files in one SWF file. Still, the expectation is
that the next version of ASP Turbine will enable the creation of one SWF file that can
play both the drawings and the sound.

6.2.3 LectureBoard is the Answer

In chapter two, the features and shortcomings of a few tools used for creating
lectures for online use were discussed. This section is devoted to show how
LectureBoard is able to solve the problems other tools suffer from.

First, and maybe most importantly, the LectureBoard solves the issue of
collaboration for lecture creation. As many users as the bandwidth allows can use the
tool and collaboratively work to build a lecture. The bandwidth limits depend on two factors: the edition and license of the Flash Communication Server MX; and the OU bandwidth limitations. The personal edition of Flash Communication Server, which is the edition used for this research, has a ceiling capacity of 1 Mb/s or 50 simultaneous connections, whichever limit is reached first. Another edition is the professional edition with a ceiling capacity of 10 Mb/s or 2,500 simultaneous connections that can be expanded up to 30 Mb/s or 7,500 simultaneous connections, whichever limit is reached first [27]. The University of Oklahoma operates at bandwidth rates of 800 kb/s and 4 Mb/s. On one hand, the maximum number of simultaneous connections of the personal will not be reached when using the OU bandwidth of 800 kb/s. On the other hand, the 4 Mb/s rate will be able to serve the 50 simultaneous connections of the personal edition. However, both the bandwidth rates of OU cannot hit the ceiling bandwidth of the professional edition with 10 Mb/s. Thus, the professional edition will be useful for the long run.

Second, the shockwave flash files are automatically generated, stored and archived on the server. This means no files are saved on the client machine or anything processed there. It also means that there is no need for Flash authoring program to build and publish Flash lectures; the Flash Turbine does the job. Third, since the LectureBoard is developed with Flash, it is platform independent, and there is no need for any special programs to be installed on the client machine, and no need to worry about lowering the security level of the web browser. Fourth, this tool accepts the input from all the input peripherals connected to the PC or the Tablet. There is no need for special pads or input devices to start using it. Fifth, the source code of the generated
lecture is still available in a text file with EMT extension. The source is kept in order to someday develop a tool to easily open and edit the lecture. And finally, the cost of running and maintaining the LectureBoard is cheap compared to other tools such as Centra Symposium. The cost of Flash Communication Server MX for educational institutes ranges from $330 for personal edition, up to $2,970 for professional one. The ASP Turbine costs around $150 for educational purposes. The rest of the application components are either provided for free such as the Internet Information Services (IIS) that runs ASP pages and comes with Windows server edition.

6.3 Targeted Platforms

The main platform targeted to create lectures with this tool was the Tablet PC. The reason why that is Tablet PC is a fully functional, pen-based computer. It also has a wide screen that is comfortable for writing and with its lightweight and wireless built-in capability it can be carried anywhere.

However, Tablet PC is a fairly new invention and not yet common in the market as notebooks. Since the LectureBoard tool was built in Flash, any other platform works fine with it. It can be used on Microsoft Windows laptops and desktops, Mac-based machines, as well as Linux-based ones. It can potentially run on Pocket PC after the screen size is adjusted.

Although this tool is suited to work with Tablet PC, there was a problem noticed while testing it. There is an obvious lag between the pen movement and the cursor response while using the tool. The lag sometimes can be as long as a second. In order to identify the reasons behind it, several tests were run on other applications. First, some
already installed drawing applications, such as Paint and Microsoft Journal, were tested for performance checking. The pen seemed to be working without noticeable lags. The next step was testing flash-based drawing boards, such as the ones described in chapter two of this thesis. The lag appears clearly, which gives an indication that the lag is the result of the slow response of Flash player on the Tablet PC system.

For the Windows-based desktop and laptop computers, the WACOM tablet was used. This tablet gave better performance on the desktops and laptops than the Tablet PC. Yet, it did not get rid of the lag completely, although it minimized it.

6.5 Development Tools

The LectureBoard used most of the tools described in chapter three. It has two parts, the client and the server parts. As for the client part, it was built completely through Flash MX, and embedded to the web through DreamWeaver. Two pre-made components were embedded in the functional interface: the voice chat and the color components. These two components are produced and freely distributed by Macromedia.

As for the server part, a number of different techniques and languages have been used. In order for the Flash interface to handle the collaboration and interaction with all the connected users, Macromedia Flash Communication Server MX is used. The recording part of the LectureBoard is handled through ASP pages written in Visual Basic scripts, COM components produced with C++ and ASP Turbine that generates the final SWF file. The lectures are archived within a database created by Microsoft Access.
Chapter Seven: Conclusion

7.1 Summary

The primary objective of this research was to use shockwave technology to build tools which can be used to enhance the online course system, eCourses. Three tools were developed for this purpose. The first tool was a website containing review material for the Electrical Engineering portion of the Fundamental of Engineering exam. The website explains theories, gives examples that demonstrates the application of the theories, and provides a few problems to examine the student’s understanding of the theories and applications. A number of tools were built using Macromedia Director to help explaining the theories. One of these tools is a conversion calculator that converts numbers from a given numbering system to the other ones. Another tool is a numerical calculator that uses both Euler and Trapezoidal methods to approximately find the value of integration, and compare it with the exact one.

The second tool is a shear-moment diagram tool that was developed to be used in Mechanics of Materials course exams. This tool enables the students in that class to earn partial credits through constructing the shear-moment diagram stated in the test problem, and submitting their answers to the server for grading. This tool was built using Macromedia Director, and designed for the eCourses system.

Finally, an online lecture creation tool, called the LectureBoard, was built and integrated with the eCourses system. The LectureBoard is a tool that was developed using Macromedia Flash MX. It enables the instructors and the students to hold an
online lecture that is provided with a drawing board, voice and text chat, and recording controls in order to record the session for later playback.

The LectureBoard has two parts: client side and server side. The communication between the client movie and the server application is possible through the Flash Communication Server MX. It operates in two modes; student mode and instructor mode. In the student mode, all the functions are made available to the user except recording. With the instructor mode, all the functions, including the recording ones, are made available. The LectureBoard was integrated with the eCourses system, and can be accessed under the administrative menu list of the instructor log-in.

7.2 Accomplishments

- A website was developed for the Fundamentals of Engineering (FE) the Electrical Engineering afternoon exams. The website contains tools such as numbering system conversion calculator and numerical integration calculator that helps the students to gain better understanding of the theory. In addition, problems were provided to help the students assess their understanding.
- A tool was developed to enable the student in a Mechanics of Materials class to submit the shear-moment diagram answers for the test problems.
- An answer-view tool for the shear-moment diagram was developed in order to enable the students to compare their submissions against the correct answers for the Mechanics of Materials class.
- A collaborative drawing board, called the LectureBoard, was developed to permit interactive and collaborative lecture creation for online use. The tool was
implemented in Flash and records the sessions in Flash format. It is also
provides voice and text chatting capabilities.

- The LectureBoard was integrated with the eCourses system, and works under
the instructor’s administration.

The tools discussed in this thesis provide solutions to some needs for online
learning. The FE website is an easy, time-saving and cost-effective way to access
review information and be prepared for the test. As for the shear-moment diagram tool,
it was developed to enable the submission of the shear-moment drawings, and improve
the eCourses system to provide the students with partial credits based on their answers.
Finally, the LectureBoard is an easy way to create web-ready lectures in collaboration
with the students on the class. Also, it provides a place for students and their instructor
to meet online for office hours to discuss some difficulties in the class material.

7.3 Recommendations for Future Research

The Tablet PC market will be growing fast, the same way the PDA grew over
the last several years. Since the Tablet PC uses the pen as the main input, it is well suit
for engineering needs. What has to be looked for is developing more tools that consider
this new platform. The LectureBoard needs more modifications and improvements to
increase performance. First, there is a need to replace the ASP pages, as well as the
COM component, with Perl pages instead. This makes it more compatible with the
eCourses system, reducing the complexity of the tool, in turn making it easy to
maintain.
So far two Flash files are needed for the playback, and that brings more complexity to the system, as well as affecting the synchronization between the drawings and the sound. These two playback files need to be combined together in one file. In order to achieve this, the new version of ASP Turbine should be used to allow the embedding of FLV file into Flash drawing file.

Finally, one of the drawbacks of the current LectureBoard is that once the lecture has been created it cannot be modified. However, since the EMT file is still stored and it contains the information for the drawings, a tool can be developed to read the EMT file, bring up the corresponding lecture structure, enable the instructor to modify and save it, and then convert the saved file to SWF again.
References

learning.html


http://www.macromedia.com/devnet/education/


[11] Nicole Ridgway, A safer place to meet (2003), FORBES.


Appendices

Appendix A: FE website sample source code

Source code for the numbering system conversion calculator

```plaintext
on DecimalToBinary num
    num = integer(num)
    bin = ""
    repeat while num <> 0
        temp = num mod 2
        bin = string(temp)&bin
        num = integer(num / 2)
    end repeat
    if bin = "" then bin = "0"
    return bin
end

on DecimalToOctal num
    num = integer(num)
    bin = ""
    repeat while num <> 0
        temp = num mod 8
        bin = string(temp)&bin
        num = integer(num / 8)
    end repeat
    if bin = "" then bin = "0"
    return bin
end

on DecimalToHexa num
    num = integer(num)
    hex = ""
    repeat while num <> 0
        temp = num mod 16
        if temp > 9 then
            hex = numToChar(charToNum("A") + temp-10)&hex
        else
            hex = string(temp)&hex
        end if
        num = integer(num / 16)
    end repeat
    if hex = "" then hex = "0"
    return hex
end

on BinaryToDecimal num
    num = String(num)
    len = num.length
```
sum = integer(0)
repeat with n = 1 to len
    sum = sum + integer(num.char[n]) * power(2, (len - n))
end repeat
return integer(sum)

on BinaryToHexa num
    bin = BinaryToDecimal (num)
    temp = DecimalToHexa (bin)

    return temp
end

on BinaryToOctal num
    --num = value(num)
    bin = BinaryToDecimal (num)
    temp = DecimalToOctal (bin)

    return temp
end

on OctalToDecimal num
    len = num.length
    sum = integer(0)
    repeat with n = 1 to len
        sum = sum + integer(num.char[n]) * power(8, (len - n))
    end repeat
    return integer(sum)
end

on OctalToBinary num
    bin = OctalToDecimal (num)
    temp = DecimalToBinary (bin)
    put temp
    return temp

end

on OctalToHexa num
    bin = OctalToDecimal (num)
    temp = DecimalToHexa (bin)
    return temp
end

on HexaToBinary num
    bin = HexaToDecimal (num)
    temp = DecimalToBinary (bin)

    return temp
end
on HexaToOctal num
    bin = HexaToBinary (num)
    temp = BinaryToOctal (bin)
    return temp
end

on HexaToDecimal num
    len = num.length
    sum = integer(0)
    repeat with n = 1 to len
        temp = num.char[n]
        if temp >= "0" and temp <= "9" then
            temp = integer (temp)
        else
            temp = integer(charToNum(temp) - charToNum("A") + 10)
        end if
        sum = sum + temp * power(16, (len - n))
    end repeat
    return integer(sum)
end

-----------------------------------------------------------------

on checkBin num
    len = num.length
    repeat with i = 1 to len
        if integer(num.char[i]) > 1 or integer(num.char[i]) < 0 then
            member("Decimal").text = "Error"
            member("binary").text = "Error"
            member("Octal").text = "Error"
            member("Hexa").text = "Error"
            return FALSE
        end if
    end repeat
    return TRUE
end

on checkDec num
    len = num.length
    repeat with i = 1 to len
        if integer(num.char[i]) > 9 or integer(num.char[i]) < 0 then
            member("Decimal").text = "Error"
            member("binary").text = "Error"
            member("Octal").text = "Error"
            member("Hexa").text = "Error"
            return FALSE
        end if
    end repeat
    return TRUE
end

on checkOct num
    len = num.length
    repeat with i = 1 to len
        if integer(num.char[i]) > 7 or integer(num.char[i]) < 0 then
member("Decimal").text = "Error"
member("Binary").text = "Error"
member("Octal").text = "Error"
member("Hexa").text = "Error"
return FALSE
end if
end repeat
return TRUE
end

on checkHex num
len = num.length
repeat with i = 1 to len
  if "0123456789ABCDEFabcdef" contains num.char[i] then
    temp = 0
  else
    member("Decimal").text = "Error"
    member("Binary").text = "Error"
    member("Octal").text = "Error"
    member("Hexa").text = "Error"
    return FALSE
  end if
  thischar = chartonum(num.char[i])
  if thischar >= chartonum("a") and thischar <= chartonum("z") then
    thischar = thischar - 32
  put numtoChar(thischar) into text.char[i]
  end if
end repeat
return TRUE
end

on Capitalize text
repeat with i = 1 to text.length
  thischar = chartonum(text.char[i])
  if thischar >= chartonum("a") and thischar <= chartonum("z") then
    thischar = thischar - 32
  put numtoChar(thischar) into text.char[i]
end if
end repeat
return text
end

on exitFrame me
  global Radix
  if Radix = 2 then
    Number1 = member("InputNum").text
    temp = checkBin(Number1)
    if temp = true then
      member("Binary").text = member("InputNum").text
      if member("InputNum").text = "" then member("Binary").text = "0"
      member("Decimal").text = String(BinaryToDecimal(Number1)) -- Works
      member("Octal").text = String(BinaryToOctal(Number1)) -- Works
      member("Hexa").text = String(BinaryToHexa(Number1)) -- Need modification
  end if
end
end if

else if Radix = 8 then
Number1 = member("InputNum").text
temp = checkOct(Number1)
if temp = true then
    member("Octal").text = member("InputNum").text
    if member("inputNum").text = "" then member("Octal").text = "0"
    member("Decimal").text = String (OctalToDecimal(Number1)) -- Working
    member("Hexa").text = String (OctalToHexa(Number1)) -- working
    member("binary").text = String (OctalToBinary(Number1)) -- Works
end if

else if Radix = 10 then
    Number1 = value (member("InputNum").text)
    temp = checkDec(Number1)
    if temp = true then
        Number1 = value (Number1)
        member("Decimal").text = member("inputNum").text
        if member("inputNum").text = "" then member("Decimal").text = "0"
        member("binary").text = String (DecimalToBinary(Number1)) -- Works
        member("Octal").text = String (DecimalToOctal(Number1)) -- Works
        member("Hexa").text = String (DecimalToHexa(Number1)) -- Works
    end if
else
    Number1 = member("InputNum").text
    temp = checkHex(Number1)
    Number1 = Capitalize(Number1)
    if temp = true then
        member("Hexa").text = member("inputNum").text
        if member("inputNum").text = "" then member("Hexa").text = "0"
        member("Decimal").text = String (HexaToDecimal(Number1))
        member("binary").text = String (HexaToBinary(Number1))
        member("Octal").text = String (HexaToOctal(Number1))
    end if
end if

    go to the frame
end
Appendix B: Mechanics of Materials test tool sample code

Parsing server problem statement

global g_BarLength -- Holds the bar length in units (inches or meters)
global g_ShearOrMoment -- Holds either "S" for shear or "M" for moment
global pnumber -- problem number, will be passed from the DB and then passed again to the grading system
global g_StudentAnswer
global g_NoOfSections

-- This step draws the problem for the student on the bar, by passing the problem's description along with the html page that shows this movie
-- Author: Mohammad Ramahi
-- Last updated: Tue, July 3, 2002.
on exitFrame me
  -- SW1 variable passed to the director holds the barlength
  if externalParamName ("sw1") = "sw1" then
    g_BarLength = integer (externalParamValue ("sw1"))
  end if

  if externalParamName ("sw3") = "sw3" then
    pnumber = externalParamValue ("sw3")
  end if

  -- The conversion factor from real units to pixels
  -- Sprite 5 is the bar graph.
  -- The below equation can do the conversion.
  if float(g_BarLength) = 0 then
    g_Barlength="1"
  end if
  Barfactor = float(sprite(5).width)/float(g_BarLength)

  -- SW2 variable passed to the director holds the graph type, if it is Shear or Moment
  -- If "S" is passed, then the Shear diagram required
  -- If "M" is passed, then Moment diagram required.
  if externalParamName ("sw2") = "sw2" then
    g_ShearOrMoment = externalParamValue ("sw2")
  end if

  -- SWList variable passed to the director holds the information needed to draw the
  -- If externalParamName ("swList") = "swList" then
  -- put externalParamValue ("swList") into Mylist --converts the text into list
  X_Axis_Length_in_Pixels = sprite(11).width -- get the screen length of the bar in pixels
  X_factor = float(X_Axis_Length_in_Pixels)/float(g_BarLength) -- conversion factor from the real units to the screen units
  LeftSideBarLocation = 105 -- the location of the beginning left side of the bar sprite
  jointcounter = 0 -- count the number of joints (5 Max)
forcecounter = 0  -- count the number of forces (5 Max)
Momentcounter = 0  -- count the number of moments (5 Max)

-- Start drawing the question diagram on the bar, by parsing the information passed from the server
-- My List holds the whole diagram information
repeat with i=1 to the number of items in Mylist
  temp = item i of Mylist

-- Different available cases
  case temp of
    "JointPin":  -- Pin Joint
      if jointcounter < 5 then  -- no more than 5 objects allowed
        sprite(95+jointcounter).member = member("pin") -- sprite 95 is the first sprite that can hold joints
        i = i+1  -- Read the location of the joint on the bar
        supportLocation = item i of Mylist
        temp = integer(supportLocation) * X_factor
        sprite(95+jointcounter).locH = LeftSideBarLocation + temp -- place the joint on place
        sprite(95+jointcounter).visible = true -- Show the joint on its place
        jointcounter = jointcounter + 1  -- increase the joint counter by 1
        member("supportloc"&string(jointcounter)).text = "("&string(integer(supportLocation))&")"
        sprite(139+jointcounter).visible = true
        sprite(139+jointcounter).loch = sprite(95+jointcounter-1).locH
      end if

    "JointRoll":  -- Roller Joint
      if jointcounter < 5 then
        sprite(95+jointcounter).member = member("roller") -- Use member "roller" to display roller joint
        i = i+1  -- Read the location of the joint on the bar
        supportLocation = item i of Mylist
        temp = integer(supportLocation) * X_factor
        sprite(95+jointcounter).locH = LeftSideBarLocation + temp
        sprite(95+jointcounter).locV =sprite(95+jointcounter).locV + 2  -- roller joint image is 2 pixels longer than the pin joint, so we need to go down 2 pixels (this is why in frame 1 script we insured the place of the pin joints to be at y = 70)
        sprite(95+jointcounter).visible = true
        jointcounter = jointcounter + 1
        member("supportloc"&string(jointcounter)).text = "("&string(integer(supportLocation))&")"
        sprite(139+jointcounter).visible = true
        sprite(139+jointcounter).loch = sprite(95+jointcounter-1).locH
      end if

    "Force":
      if item i+2 of Mylist > 0 then
        if forcecounter < 5 then  -- no more than 5 forces/distributed forces
          sprite(105+forcecounter).member = member("PointForce") -- sprite 105 is the first sprite that can hold forces
          i = i+1  -- Read the location of the force on the bar
          forceLocation = integer(item i of Mylist)
          temp = forceLocation * X_factor -- convert the location from real world units to the screen pixels
sprite(105+forcecounter).locH = LeftSideBarLocation + temp -- Place the force on place
sprite(105+forcecounter).rotation = 360 -- Keep the same force graph (pointing up)
sprite(105+forcecounter).visible = true -- Show the force
forcecounter = forcecounter + 1 -- increase the force counter
-- Reading and showing the magnitude of the force
i = i + 1 -- Read the magnitude of the force
forceMagnitude = integer(item i of Mylist)
temp = "forcemag"&string(forcecounter)
member(temp).text = ("&string(forceLocation)&","&string(forceMagnitude)&")
sprite(119+forcecounter).visible = true
sprite(119+forcecounter).locH = sprite(105+forcecounter-1).locH
end if
else
-- Same comments for previous code apply, differences are stated below
if forcecounter < 5 then
sprite(105+forcecounter).member = member("PointForce")
i = i + 1
forceLocation = integer(item i of Mylist)
temp = forceLocation * X_factor
sprite(105+forcecounter).locH = LeftSideBarLocation + temp
sprite(105+forcecounter).rotation = 180 -- Rotate the force image in order to make is pointing down
sprite(105+forcecounter).visible = true
forcecounter = forcecounter + 1
-- Reading and showing the magnitude of the force
i = i + 1 -- Read the magnitude of the force
forceMagnitude = -1*integer(item i of Mylist)
temp = "forcemag"&string(forcecounter)
member(temp).text = ("&string(forceLocation)&","&string(forceMagnitude)&")
sprite(119+forcecounter).visible = true
sprite(119+forcecounter).locH = sprite(105+forcecounter-1).locH
end if
end if

"DistRect":
if item i+2 of Mylist > 0 then
if forcecounter < 5 then
sprite(105+forcecounter).member = member("druforce") -- Still using sprite 105
i = i + 1 -- Read the starting location of the distributed force
forceLocation = integer(item i of Mylist)
temp = forceLocation * X_factor
-- Reading the magnitude of the force
i = i + 1 -- Read the magnitude of the force
forceMagnitude = integer(item i of Mylist)
i = i + 1 -- Read the Width of the distributed force
forceWidthReal = integer(item i of Mylist)
forceWidth = forceWidthReal * X_factor -- convert the width units into pixels
sprite(105+forcecounter).width = forcewidth -- expand the force sprite width to the D-force width
sprite(105+forcecounter).locH = LeftSideBarLocation + temp + forcewidth/2 -- locate the D-force on the bar, the starting point is the left edge of the D-force
sprite(105+forcecounter).visible = true
forcecounter = forcecounter + 1

-- Displaying the magnitude of the force (uses two force spaces because two locations for the points)
    temp = "forcemag"&string(forcecounter)
    member(temp).text = "("&string(forceLocation+forceWidthReal)&","&string(forceMagnitude)&")"
    sprite(119+forcecounter).visible = true
    sprite(119+forcecounter).loch = sprite(105+forcecounter-1).loch + forcewidth/2 -
    sprite(119+forcecounter).width/2
    forcecounter = forcecounter + 1
    temp = "forcemag"&string(forcecounter)
    member(temp).text = "("&string(forceLocation)&","&string(forceMagnitude)&")"
    sprite(119+forcecounter).visible = true
    sprite(119+forcecounter).loch = sprite(105+forcecounter-2).loch - forcewidth/2
end if
else
    -- Same comments for previous code apply, differences are stated below
    if forcecounter < 5 then
        sprite(105+forcecounter).member = member("drdforce") -- Use the DForce rectangular illustration
        i = i+1
        forceLocation = integer(item i of Mylist)
        temp = forceLocation * X_factor
        -- Reading the magnitude of the force
        i = i+1
        forceMagnitude = -1 * integer (item i of Mylist)
        i = i+1
        forceWidthReal = integer(item i of Mylist)
        forcewidth = forceWidthReal * X_factor -- convert the width units into pixels
        sprite(105+forcecounter).width = forcewidth
        sprite(105+forcecounter).loch = LeftSideBarLocation + temp + forcewidth/2
        sprite(105+forcecounter).visible = true
        forcecounter = forcecounter + 1

        -- Displaying the magnitude of the force
        temp = "forcemag"&string(forcecounter)
        member(temp).text = "("&string(forceLocation+forceWidthReal)&","&string(forceMagnitude)&")"
        sprite(119+forcecounter).visible = true
        sprite(119+forcecounter).loch = sprite(105+forcecounter-1).loch + forcewidth/2 -
        sprite(119+forcecounter).width/2
        forcecounter = forcecounter + 1
        temp = "forcemag"&string(forcecounter)
        member(temp).text = "("&string(forceLocation)&","&string(forceMagnitude)&")"
        sprite(119+forcecounter).visible = true
        sprite(119+forcecounter).loch = sprite(105+forcecounter-2).loch - forcewidth/2
    end if
end if

"DistTriL":
if item i+2 of Mylist > 0 then
if forcecounter < 5 then
    sprite(105+forcecounter).member = member("dtuforce")
    i = i+1 -- Read the location
    forceLocation = integer(item i of Mylist)
    temp = forceLocation * X_factor
    -- Reading the magnitude of the force
    i = i+1 -- Read the Magnitude
    forceMagnitude = integer (item i of Mylist)

    i = i+1 -- Read the width
    forceWidthReal = integer(item i of Mylist)
    forcerwidth = forceWidthReal * X_factor -- conver the width units into pixels

    sprite(105+forcecounter).width = forcerwidth
    sprite(105+forcecounter).fliph = false -- Use the default flipping for this D-Force shape
    sprite(105+forcecounter).loch = LeftSideBarLocation + temp + forcerwidth/2
    sprite(105+forcecounter).visible = true
    forcecounter = forcecounter + 1

    -- Displaying the magnitude of the force
    temp = "forcemag"&string(forcecounter)
    member(temp).text = "("&string(forceLocation+forceWidthReal)&","&string(0)&")"
    sprite(119+forcecounter).loch = sprite(105+forcecounter-1).loch + forcerwidth/2 -
    sprite(119+forcecounter).width/2
    forcecounter = forcecounter + 1

    temp = "forcemag"&string(forcecounter)
    member(temp).text = "("&string(forceLocation)&","&string(forceMagnitude)&")"
    sprite(119+forcecounter).visible = true
    sprite(119+forcecounter).loch = sprite(105+forcecounter-2).loch - forcerwidth/2

end if

else
    -- Same comments for previous code apply, differences are stated below
    if forcecounter < 5 then
        sprite(105+forcecounter).member = member("dtdforce")
        i = i+1
        forceLocation = integer(item i of Mylist)
        temp = forceLocation * X_factor
        -- Reading the magnitude of the force
        i = i+1
        forceMagnitude = -1 * integer (item i of Mylist)

        i = i+1
        forceWidthReal = integer(item i of Mylist)
        forcerwidth = forceWidthReal * X_factor -- conver the width units into pixels
        sprite(105+forcecounter).width = forcerwidth
        sprite(105+forcecounter).fliph = false
        sprite(105+forcecounter).loch = LeftSideBarLocation + temp + forcerwidth/2
        sprite(105+forcecounter).visible = true
        forcecounter = forcecounter + 1

        -- Displaying the magnitude of the force
        temp = "forcemag"&string(forcecounter)
member(temp).text = "("&string(forceLocation+forceWidthReal)&","&string(0)&")"  
sprite(119+forcecounter).visible = true  
sprite(119+forcecounter).loch = sprite(105+forcecounter-1).loch + forcewidth/2 -  
sprite(119+forcecounter).width/2  
forcecounter = forcecounter + 1  
temp = "forcemag"&string(forcecounter)  
member(temp).text = "("&string(forceLocation)&","&string(forceMagnitude)&")"  
sprite(119+forcecounter).visible = true  
sprite(119+forcecounter).loch = sprite(105+forcecounter-2).loch - forcewidth/2  
end if  
end if

"DistTriR":  
if item i+2 of Mylist > 0 then  
  -- Same comments for previous code apply, differences are stated below  
  if forcecounter < 5 then  
    sprite(105+forcecounter).member = member("dtuforce")  
    i = i+1  
    forceLocation = integer(item i of Mylist)  
    temp = forceLocation * X_factor  
    -- Reading the magnitude of the force  
    i = i+1  
    forceMagnitude = integer(item i of Mylist)  
    i = i+1  
    forceWidthReal = integer(item i of Mylist)  
    forcewidth = forceWidthReal * X_factor -- conver the width units into pixels  
    sprite(105+forcecounter).width = forcewidth  
    sprite(105+forcecounter).fliph = true -- flip the D-force image horizontally to get the 
    required shape  
    sprite(105+forcecounter).loch = LeftSideBarLocation + temp + forcewidth/2  
    sprite(105+forcecounter).visible = true  
    forcecounter = forcecounter + 1  
    -- Displaying the magnitude of the force  
    temp = "forcemag"&string(forcecounter)  
    member(temp).text =  
    "("&string(forceLocation+forceWidthReal)&","&string(forceMagnitude)&")"  
    sprite(119+forcecounter).visible = true  
    sprite(119+forcecounter).loch = sprite(105+forcecounter-1).loch + forcewidth/2 -  
    sprite(119+forcecounter).width/2  
    forcecounter = forcecounter + 1  
    temp = "forcemag"&string(forcecounter)  
    member(temp).text = "("&string(forceLocation)&","&string(0)&")"  
    sprite(119+forcecounter).visible = true  
    sprite(119+forcecounter).loch = sprite(105+forcecounter-2).loch - forcewidth/2  
  end if  
else  
  -- Same comments for previous code apply, differences are stated below  
  if forcecounter < 5 then  
    sprite(105+forcecounter).member = member("dtdforce")  
    i = i+1  
  end if
forceLocation = integer(item i of Mylist)
temp = forceLocation * X_factor
-- Reading the magnitude of the force
i = i+1
forceMagnitude = -1*integer (item i of Mylist)

i = i+1
forceWidthReal = integer(item i of Mylist)
forcewidth = forceWidthReal * X_factor -- convert the width units into pixels

sprite(105+forcecounter).width = forcewidth
sprite(105+forcecounter).flipH = true
sprite(105+forcecounter).locH = LeftSideBarLocation + temp + forcewidth/2
sprite(105+forcecounter).visible = true
forcecounter = forcecounter + 1

-- Displaying the magnitude of the force
temp = "forcemag"&string(forcecounter)
member(temp).text = "("&string(forceLocation+forceWidthReal)&","&string(forceMagnitude)&")"
sprite(119+forcecounter).visible = true
sprite(119+forcecounter).locH = sprite(105+forcecounter-1).locH + forcewidth/2 -
sprite(119+forcecounter).width/2
forcecounter = forcecounter + 1

end if
end if

"Moment":
if item i+2 of Mylist < 0 then
if momentcounter < 5 then
sprite(100+momentcounter).member = member("moment") -- sprite 100 is the first
sprite that can hold moments
i = i+1 -- Read the location
momentLocation = integer(item i of Mylist)
temp = momentLocation * X_factor -- convert the location from the actual units to pixels
sprite(100+momentcounter).locH = LeftSideBarLocation + temp -- Place the moment
sprite(100+momentcounter).flipH = true -- The moment image flipped to be shown as
Clockwise rotation
sprite(100+momentcounter).visible = true
momentcounter = momentcounter + 1

-- Reading and showing the magnitude of the moment
i = i + 1 -- Read the magnitude of the force
MomentMagnitude = -1*integer (item i of Mylist)
temp = "momentmag"&string(momentcounter)
member(temp).text = "("&string(momentLocation)&","&string(MomentMagnitude)&")"
sprite(124+momentcounter).visible = true
sprite(124+momentcounter).locH = sprite(100+momentcounter-1).locH

end if
else
-- Same comments for previous code apply, differences are stated below
if momentcounter < 5 then
    sprite(100+momentcounter).member = member("moment")
i = i+1
momentLocation = integer(item i of Mylist)
temp = momentLocation * X_factor -- convert the location from the actual units to pixels
sprite(100+momentcounter).locH = LeftSideBarLocation + temp
sprite(100+momentcounter).flipH = false -- the default image shows counter-clockwise rotation
sprite(100+momentcounter).visible = true
momentcounter = momentcounter + 1

-- Reading and showing the magnitude of the moment
i = i + 1 -- Read the magnitude of the force
MomentMagnitude = integer (item i of Mylist)
temp = "momentmag"&string(momentcounter)
member(temp).text = "("&string(momentLocation)&","&string(MomentMagnitude)&")"
sprite(124+momentcounter).visible = true
sprite(124+momentcounter).loch = sprite(100+momentcounter-1).locH + sprite(100+momentcounter-1).width
end if
end if

otherwise: -- for any unknown command, just skip it
    i = i+1
end case
end repeat
end

if externalParamName ("sw4") = "sw4" then
    g_StudentAnswer = externalParamValue ("sw4")
g_NoOfSections = integer(item 1 of g_StudentAnswer)
end if

go to frame 6
end
Appendix C: LectureBoard sample code

Source code for the drawings

```javascript
_root.drag_tools = false;
eraser_thickness_factor = 10;
function drawRect(clip, x1, y1, x2, y2, thikness, linecolor, alpha) {
    _root.clip.lineStyle(thikness, linecolor, alpha);
    _root.clip.moveTo(x2, y2);
    _root.clip.lineTo(x1, y2);
    _root.clip.lineTo(x1, y1);
    _root.clip.lineTo(x2, y1);
    _root.clip.lineTo(x2, y2);
}
function drawLine(clip, x1, y1, x2, y2, thikness, linecolor, alpha) {
    _root.clip.lineStyle(thikness, linecolor, alpha);
    _root.clip.moveTo(x1, y1);
    _root.clip.lineTo(x2, y2);
}
function drawCircle(clip, x1, y1, x2, y2, thikness, linecolor, alpha) {
    _root.clip.lineStyle(thikness, linecolor, alpha);
    r = Math.sqrt(Math.pow(x1-x2, 2)+Math.pow(y1-y2, 2));
    if ((x1+r)>clipBounds.xMax || (x1-r)<clipBounds.xMin) {
        return -1;
    } else {
        if ((y1+r)>clipBounds.yMax || (y1-r)<clipBounds.yMin) {
            return -1;
        }
    }
    _root.clip.moveTo(x1-r, y1);
    a = r*0.4086; // Radius multiplied by constant
    b = r*0.7071; // Radius multiplied by constant
    _root.clip.curveTo(x1-r, y1-a, x1-b, y1-b);
    _root.clip.curveTo(x1-a, y1-r, x1, y1-r);
    _root.clip.curveTo(x1+a, y1-r, x1+b, y1-b);
    _root.clip.curveTo(x1+r, y1+a, x1+b, y1+b);
    _root.clip.curveTo(x1+a, y1+r, x1, y1+r);
    _root.clip.curveTo(x1-a, y1+r, x1-b, y1+b);
    _root.clip.curveTo(x1-r, y1+a, x1-r, y1);
}
function polarConversion(r, t) {
    var x = r*Math.cos(t*(Math.PI/180));
    var y = r*Math.sin(t*(Math.PI/180));
    return {x:x, y:y};
}
function drawArrowT(clip, x1, y1, x2, y2, thikness, linecolor, alpha) {
    // the line below determines the thickness, color and transperancy of the line
    _root.clip.lineStyle(thikness, linecolor, alpha);
    // find the angel
    var angel;
```

75
angel = Math.atan2((y2-y1), (x2-x1))*(180/Math.PI);
// finds the arctan (which is the angle of slope)
var offsetA = polarConversion(10, angel + 157.5);
var offsetB = polarConversion(10, angel - 157.5);
_root.clip.moveTo(x2, y2);
_root.clip.lineTo(x2 + offsetB.x, y2 + offsetB.y);
// draws one of the sides of the arrow
_root.clip.moveTo(x2, y2);
_root.clip.lineTo(x2 + offsetA.x, y2 + offsetA.y);
// draws the other side of the arrow.
}

function changeThickness() {
    _global.lineThickness = thickness_mc.getSelectedIndex() + 1;
}

//-----------------------------------------
someListener = new Object();
someListener.onKeyUp = function() {
    if (Key.getCode() == 46) {
        var temp = "clip" + _global.selectedClip;
        trace(temp);
        client_nc.call("deleteClip", null, temp, _global.selectedClip);
        _root.clip.removeMovieClip();
        _global.selectedClip = 0;
    }
};
Key.addListener(someListener);

_root.onMouseUp = function() {
    if (startDrawing == true && _root.drag_tools == false) {
        startDrawing = false;
        clipBounds = drawarea_mc.getBounds(_root);
        if (_xmouse > clipBounds.xMax || _xmouse < clipBounds.xMin) {
            return;
        }
        if (_ymouse > clipBounds.yMax || _ymouse < clipBounds.yMin) {
            return;
        }
        var pointsArray = new Array();
        pointsArray[0] = _global.x1;
        pointsArray[1] = _global.y1;
        pointsArray[2] = _xmouse;
        pointsArray[3] = _ymouse;
        thickness = _global.lineThickness;
        _root.clip.moveTo(_global.x1, _global.y1);
        switch (shape) {
            case "rect":
                drawRect(clip, _global.x1, _global.y1, _xmouse, _ymouse,
                         _global.lineThickness, gFlashCom.userprefs.color, 100);
                var myshape = "rect";
                break;
            case "line":
                drawLine(clip, _global.x1, _global.y1, _xmouse, _ymouse,
                         _global.lineThickness, gFlashCom.userprefs.color, 100);
                var myshape = "line";
                break;
        }
    }
}
break;
case "circle":
    var checkflag = 0;
    checkflag = drawCircle(clip, _global.x1, _global.y1, _xmouse, _ymouse,
    _global.lineThickness, gFlashCom.userprefs.color, 100);
    if (checkflag == -1) {
        return;
    }
    var myshape = "circle";
break;
case "arrowt":
    drawLine(clip, _global.x1, _global.y1, _xmouse, _ymouse,
    _global.lineThickness, gFlashCom.userprefs.color, 100);
    drawArrowT(clip, _global.x1, _global.y1, _xmouse, _ymouse,
    _global.lineThickness, gFlashCom.userprefs.color, 100);
    var myshape = "arrowt";
break;
case "freeh":
    var myshape = "freeh";
    xArray.push(_xmouse);
    yArray.push(_ymouse);
    for (var i = 0; i<xArray.length; i += 2) {
        pointsArray[2*i] = xArray[i];
        pointsArray[2*i+1] = yArray[i];
    }
    delete xArray;
    delete yArray;
break;
case "eraser":
    var myshape = "eraser";
    xArray.push(_xmouse);
    yArray.push(_ymouse);
    for (var i = 0; i<xArray.length; i += 2) {
        pointsArray[2*i] = xArray[i];
        pointsArray[2*i+1] = yArray[i];
    }
    delete xArray;
    delete yArray;
break;
}
var clipcolor = gFlashCom.userprefs.color;
if (shape == "eraser") {
    var dataArray = {lineColor:"0xFFFFFF", lineThickness:thickness,
    lineDepth:0, lineAlpha:100, linePoints:pointsArray};
} else {
    var dataArray = {lineColor:clipcolor, lineThickness:thickness,
    lineDepth:0, lineAlpha:100, linePoints:pointsArray};
}
var clientMessage = {user_:userName, command_:"draw", shape_:myshape,
data_:dataArray};
client_nc.call("drawingRequestFromClient", null, clientMessage);
_root.clip.removeMovieClip();
} else {
if (shape == "sel") {
    _root.createEmptyMovieClip("clip", 3);
for (var i = 0; i<NumOfLines; i++) {
    temp = "clip"+(10+i);
    if (_root[temp].hitTest(_xmouse, _ymouse, false)) {
        _global.selectedClip = 10+i;
        objectBounds = _root[temp].getBounds(_root);
        drawRect(_clip, objectBounds.xMax, objectBounds.yMax, objectBounds.xMin, objectBounds.yMin, 1, 0xffff00, 100);
        break;
    }
}

_root.onMouseMove = function() {
    if (startDrawing == true && _root.drag_tools == false) {
        _root.clip.removeMovieClip();
        _root.createEmptyMovieClip("clip", 3);
        clip.swapDepths(_global.lineDepth+1);
        clipBounds = drawarea_mc.getBounds(_root);
        if (_xmouse>clipBounds.xMax || _xmouse<clipBounds.xMin) {
            return;
        }
        if (_ymouse>clipBounds.yMax || _ymouse<clipBounds.yMin) {
            return;
        }
        switch (shape) {
            case "rect":
                drawRect(_clip, _global.x1, _global.y1, _xmouse, _ymouse, _global.lineThickness, gFlashCom.userprefs.color, 100);
                break;
            case "line":
                drawLine(_clip, _global.x1, _global.y1, _xmouse, _ymouse, _global.lineThickness, gFlashCom.userprefs.color, 100);
                break;
            case "circle":
                drawCircle(_clip, _global.x1, _global.y1, _xmouse, _ymouse, _global.lineThickness, gFlashCom.userprefs.color, 100);
                break;
            case "arrowt":
                drawLine(_clip, _global.x1, _global.y1, _xmouse, _ymouse, _global.lineThickness, gFlashCom.userprefs.color, 100);
                drawArrowT(_clip, _global.x1, _global.y1, _xmouse, _ymouse, _global.lineThickness, gFlashCom.userprefs.color, 100);
                break;
            case "freeh":
                xArray.push(_xmouse);
                yArray.push(_ymouse);
                _root.clip.lineStyle(_global.lineThickness, gFlashCom.userprefs.color, 100);
                clip.moveTo(_global.x1, _global.y1);
                for (var i = 0; i<xArray.length; i++) {
                    clip.lineTo(xArray[i], yArray[i]);
                }
                break;
        }
    }
}
case "eraser":
    //clip.swapDepths(_global.lineDepth+1);
    xArray.push(_xmouse);
    yArray.push(_ymouse);
    _root.clip.lineStyle(_global.lineThickness*eraser_thickness_factor,
"0xFFFFFFFF", 100);
    clip.moveTo(_global.x1, _global.y1);
    for (var i = 0; i<xArray.length; i++) {
        clip.lineTo(xArray[i], yArray[i]);
    }
    break;

_root.clip.updateAfterEvent();

};
_root.onMouseDown = function() {
    if (shape != "sel" && shape != "") {
        if (drawarea_mc.hitTest(_xmouse, _ymouse, false)) {
            _root.drag_tools = false;
            startDrawing = true;
            _global.x1 = _xmouse;
            _global.y1 = _ymouse;
            _root.createEmptyMovieClip("clip", 3);// level 1 for grids
            _root.clip.lineStyle(_global.lineThickness, gFlashCom.userprefs.color,
100);
            xArray = new Array();
            yArray = new Array();
            xArray.push(_xmouse);
            yArray.push(_ymouse);
        }
    }
};

function doSend() {
    if (length(_root.Message.text)>0) {
        _root.client_nc.call("msgFromClient", null, _root.Message.text);
    }
    _root.Message.text = "";
}